SSRG International Journal of Electronics and Communication Engineering
ISSN: 2348-8549/ https://doi.org/10.14445/23488549/IJECE-V12I5P104

Volume 12 Issue 5, 33-49, May 2025


*Original Article*

# Development of Relevance Propagation Rule-based Systems and Faster Mask R-CNN for IoT-Enabled Surveillance Drones to Enhance Autonomous Navigation and Collision Avoidance

L. Ganesh Babu[1], V. Balambica[2], M. Achudhan[3]

[1,2,3]*Department of Mechatronics, Bharath Institute of Higher Education and Research, Chennai, Tamilnadu, India.*

[1]*Corresponding Author : ganeshme68@gmail.com*

Received: 03 March 2025 Revised: 05 April 2025 Accepted: 05 May 2025 Published: 27 May 2025

***Abstract*** - *The study focuses on enhancing autonomous navigation and improving collision avoidance by developing a Relevance Propagation Rule-Based on Faster Mask R-CNN (RPR-FMRCNN) leveraging IoT-enabled surveillance drones to enhance smart decision-making. Autonomous drones are increasingly being used for various tasks, but operating them safely and efficiently in complex environments remains a significant challenge, particularly in avoiding collisions with moving obstacles. Existing approaches struggle with real-time decision-making and are often imprecise and unstable in rapidly changing scenarios. The paper proposes a hybrid system for obstacle recognition to address these issues. IoT is used to gather real-time information, ensuring immediate updates. The rule-based approach prioritizes obstacle relevance to dynamically create safer paths, while FMRCNN accurately identifies obstacles, providing boundaries and segmentation masks for each object. IoT-enabled surveillance drones offer seamless connectivity and data exchange, facilitating continuous environmental updates and informed decision-making. The primary goal is to develop a system capable of autonomously determining optimal navigation routes, accurately identifying and categorizing obstacles, and making well-informed decisions to avoid them. By combining deep neural networks with IoT, the model aims to provide real-time processing with improved precision and efficiency. The study's results show significant improvements in obstacle recognition and navigation, with the system performing better in dynamic environments. The outcomes demonstrate a substantial reduction in crashes, enhancing the overall reliability of drones. Compared to existing methods, the proposed model improves collision avoidance efficiency by 15% and navigation accuracy by 12%, signaling promising advancements in autonomous drone systems for various applications.*

***Keywords*** - *Relevance Propagation Rule-Based Systems, Faster Mask R-CNN, IoT-Enabled surveillance drones, Autonomous navigation, Collision avoidance, Intelligent Decision-Making, Obstacle detection, Real-Time data processing, Deep learning, Drone navigation.*

## 1. Introduction

Unmanned Aerial Vehicles (UAVs) are widely employed across various fields, including healthcare, military missions, logistics, target tracking, monitoring, surveillance, and communications. Their full potential, particularly for quadcopters, is limited by battery life. Most quadcopters currently operate for only 15 to 40 minutes on a single battery charge [1]. Due to this limited battery life, a quadcopter must land, return to a fixed point, and either recharge or change its batteries before resuming operation. This process often requires human intervention, which leads to increased personnel costs and operating expenses. Autonomous wireless charging for quadcopters is therefore crucial to enabling continuous flight operations and overcoming these limitations [2]. One of the primary challenges hindering continuous UAV operations is battery life. To address this, scientists have developed an external contact recharge mechanism. In the surface contact recharging method, the UAV must align correctly with the ground to recharge. Using metal connectors for UAV battery recharging is unsafe in varying weather conditions, such as rain [3]. While UAV batteries can be recharged at the launch site or swapped out, swapping batteries is not ideal because the process may differ depending on the UAV model. Wireless charging is the most suitable, secure, and controllable solution, especially for cases where the UAV does not land precisely on the charging pad [4]. In the Wireless Power Transfer (WPT) approach, charging pads are typically deployed in different areas. When a quadcopter detects that its battery is low, it uses GPS coordinates to locate the nearest charging station and recharge. GPS-based landing can be unreliable, potentially preventing precise touchdown [5]. Vision-based assistance can solve this issue, helping the UAV approach the charging pad accurately and providing additional information about its location. Robotics involves the development, construction, and operation of robots to perform specific tasks either autonomously or with human guidance, which is closely linked with these advancements [6]. Robots are equipped with various sensors, including lidar and cameras, that generate large amounts of visual data. This data can be used to assess the robot's tasks, environment, and movement. The collection, storage, processing, and evaluation of visual data, such as images and videos, fall under the category of visual management of information. This data can be vast, complex, and unstructured, making it difficult to handle and analyze [7]. Visual management of information methods and tools is used to extract valuable insights from visual data, making it accessible for both machines and humans. This is crucial for autonomous robot operations, where decisions must be made based on visual input. For example, robots can use visual data to detect people and objects, navigate complex environments, and perform manipulation and grasping tasks [8]. At the same time, robotics is driving advancements in the visual management of information by producing large amounts of visual data that require efficient processing and analysis in real time. Thus, the visual management of information and robotics are closely related fields, each advancing the other. Improvements in one area drive progress in the other, leading to the development of autonomous robotics and intelligent technologies. These UAVs can be used for various tasks, such as package delivery and specialized underwater operations [9].

The Internet of Things (IoT) seeks to connect devices anywhere, at any time, over any network, and with any type of service. This concept enables UAVs to become an essential component of the IoT ecosystem. A UAV's body can be considered its physical entity, while its controller represents its virtual entity. Intelligent UAV management systems allow remote control of multiple UAVs from any device, anywhere. While drones are often designed for specific tasks, such as delivering mail, they can also provide value-added services related to IoT [6]. For example, while monitoring air pollution, a drone could also provide real-time traffic conditions for specific streets, benefiting stakeholders like the Transportation Security Administration.

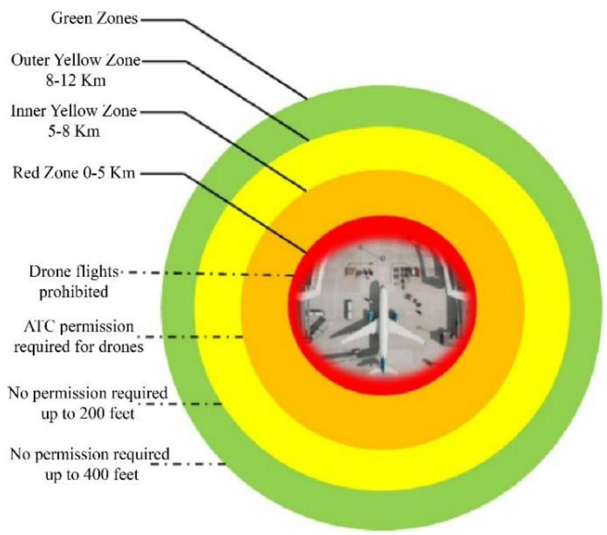



**Fig. 1 Flying zones identified in the drone rules**

This multi-purpose use of drones could generate additional income for their owners without requiring separate investments from other stakeholders [10]. Continuously using Machine-Type Communication (MTC) modules and maintaining constant internet connectivity can overload the network and drain the drone's battery faster. Therefore, finding a balance between power consumption and efficient network usage is key to maintaining UAV performance [11].

A Unique Identification Number (UIN) should be assigned to every drone, while *Digital Sky* serves as a nationwide Unmanned Traffic Management (UTM) platform that streamlines the certification process for drone ownership and operation. The primary functionalities of the Digital Sky application include identification, flight planning, and reporting. Indian airspace is divided into three zones: Red, Yellow, and Green. Green zones are classified as Unrestricted and Unregulated Airspace globally [12]. Figure 1 visually represents these zones. Power consumption is a major concern for battery-powered devices, and this is especially critical for drones, which rely entirely on batteries for operation and flight. For example, a drone might be programmed to deliver two packages to two different locations [13]. A challenge arises when a drone deviates from its intended course, possibly due to adverse weather conditions. In such cases, extra precautions must be taken to avoid collisions with other drones in the air. When selecting a drone or a fleet of drones to carry out a specific task or value-added service, several factors must be considered. These considerations not only help provide additional services but also ensure the safety of the drones by preventing collisions [14].

## 2. Related Works

UAVs are among the numerous autonomous systems where automatic obstacle avoidance is crucial. In recent years, the development of UAVs has focused significantly on creating smart and independent quadrotors due to their potential to enhance security and intuitive control across various applications, such as traffic monitoring, delivery, building surveillance, and mapping [15].

**Fig. 2 UAV work for an environmental condition**

To achieve these objectives and adapt to varying operating conditions, a UAV control system must simultaneously manage vision, control, and positioning. For optimal functionality in social environments typically unstructured and highly dynamic, a UAV must communicate effectively with other active entities, such as vehicles and pedestrians (as illustrated in Figure 2). The quadrotor must sense and quickly react to its surroundings to address the challenge of obstacle avoidance [16]. A deep learning-based method was introduced, which was rapidly adapted to an approximation map. Developed a UAV control algorithm that integrates data collected over time, potentially improving decision-making accuracy. While supervised learning techniques offer a more practical approach to acquiring efficient flight rules, they face the challenge of obtaining sufficient expert trajectories for replication [17]. Human pilots are required to create collision-avoidance trajectories to teach the robotic platform how to respond in hazardous situations. When applying deep learning to UAV tasks, tracking accuracy and processing complexity are key considerations. The obstacle avoidance challenge aims to optimize accuracy while operating within limited computing resources enabled by specialized technology [18]. Most current intrusion detection techniques for the IoT are designed to detect Denial of Service (DoS) or routing attacks. Additional research has also focused on identifying unauthorized access to memory in low-power IoT systems. Relies on lightweight specification-based intrusion detection to identify misbehavior in any IoT device integrated into a Cyber-Physical System (CPS) [19].

A major focus of autonomous quadrotor studies that often utilize GPS is obstacle avoidance. GPS can suffer from signal loss in both indoor and outdoor environments. By integrating data from the UAV's other sensors, its precise location can still be determined with minimal processing burden. Deep learning has been increasingly applied to UAV obstacle avoidance, leading to extensive research on developing deep neural network-based learning schemes using raw sensory data [20]. An appropriate network is needed for UAVs to respond quickly and accurately to their operating environments and make real-time decisions during flight. To facilitate autonomous obstacle avoidance, a lightweight model is essential. MobileNets utilized resolution multipliers, width multipliers, and depth-wise separable convolutions to reduce latency and model size [21]. Shufflenet-V2, following four principles, designed a network architecture that proved to be faster and as accurate as MobileNet-V2. Despite their advantages, these models still incur significant computational costs when applied to UAV obstacle avoidance. Some processing can be reduced by recognizing that lower-level blocks can still generate accurate results for UAV manoeuvring [22].

One common approach to obstacle avoidance involves using specific control algorithms to prevent robot collisions in a dynamic environment. This includes accounting for dynamic elements, such as pedestrians, which act as moving barriers. To create dynamic barriers, complex models are required, along with actions based on the robot's potential states, to avoid collisions. It is difficult to account for every possible situation in real-world scenarios, as dynamic obstacles can move unpredictably. When faced with an unaccounted scenario, the system may lose control [23]. To address this, various studies have explored learning-based techniques. Unfortunately, numerous models are required to account for the diverse and unpredictable factors in dynamic environments, as the training process relies on a well-defined framework for dynamic barriers. In this research, datasets containing images with dynamic elements will be used to build a computational model [24]. Using a generated depth map, a deterministic arbitration mechanism will be applied to control the UAV's angle across two rotational Degrees of Freedom (DOF), guiding it away from obstacles. The CNN, trained on samples containing dynamic elements, shares a feature extractor with the regression stream. The primary objective is to find a clear escape route in the UAV's surroundings. The steering angles from autonomous driving datasets are used as labels in the regression model training, and datasets in the classification branch are labeled as positive or negative based on the proximity of nearby vehicles and objects [25].

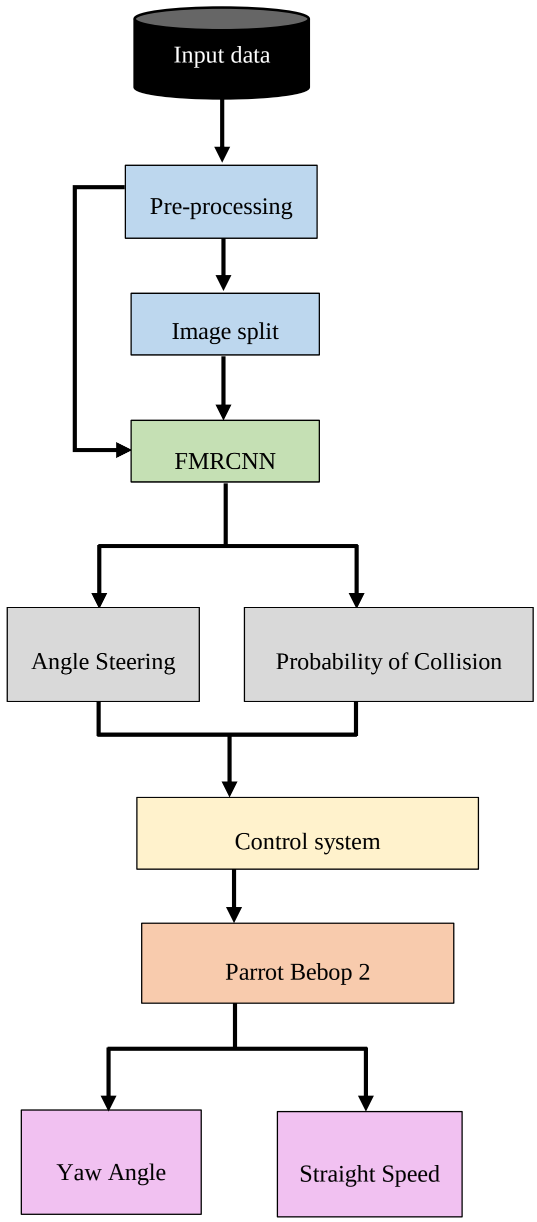



**Fig. 3 Proposed architecture**

## 3. Materials and Methods

The goal of the research on the significance of RPR-FMRCNN is to significantly enhance self-navigation and collision avoidance in complex environments, particularly for IoT-enabled surveillance drones, as shown in Figure 3. Dynamic obstacle identification, safe navigation, and decision-making pose major challenges for autonomous drones, which are widely used in delivery, security, and monitoring applications. Traditional approaches often struggle with real-time processing and lack flexibility in rapidly changing environments, increasing the risk of crashes. This research addresses these issues by prioritizing and assessing the significance of detected obstacles, combining RPR-FMRCNN, a powerful deep-learning model for object detection and segmentation. The integration of IoT technology allows drones to continuously gather real-time environmental data, ensuring prompt and accurate decision-making during flight. By leveraging these technologies, the system can more effectively identify, classify, and avoid obstacles while dynamically generating routes that minimize collision risks. The proposed approach greatly enhances the drone's ability to navigate autonomously and reliably in challenging environments. It improves obstacle recognition and decision-making capabilities, ensuring safer and more efficient drone operations in real-time scenarios. This advancement is crucial for applications such as disaster response, infrastructure monitoring, and surveillance.

In this study, researchers avoid obstacles by using inexpensive and effective networks that meet the aforementioned conditions. To accomplish this, it is an extremely dynamic situation, such as one where there are a lot of pedestrians. Figure 4 displays the coordinate system and model of the UAV. The sets of data labeled by collision probability are used to train an FMECNN model policy. Two operational states have the same network topology even though they have different regulations. Thus, the additional calculation is negligible. The rest of this section will provide a detailed introduction to each stage.

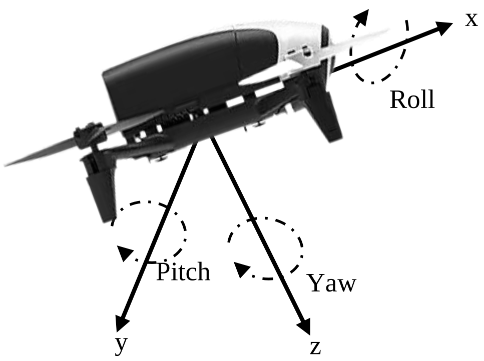



**Fig. 4 UAV and its coordinate system**

### *3.1. Problem Definition*

The UAV must discover an obstacle-free path in an area with many obstacles, as seen in Figure 5, where the starting point, destination, and impediment locations are randomly generated. Among these, the green rectangle frame denotes the UAV's onboard camera's range of vision, and the red regions indicate no-fly zones caused by challenges, and the yellow star indicates the intended location. When the UAV approaches an obstacle, it should determine how to safely navigate through a hazardous region by evaluating its surroundings and the goal location. This includes determining whether to alter its heading or flight path angle. The UAV must then carry out its duties in a continuing manner. The mission resumes when the UAV runs into an obstruction or arrives at its objective.

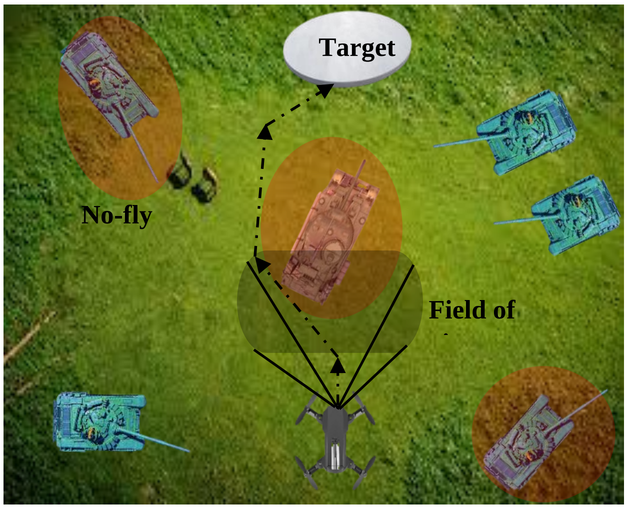



**Fig. 5 Diagram of mission scenario**

The key problem addressed in this research is the enhancement of autonomous drone navigation and collision avoidance by utilizing loT-enabled data, advanced object detection through FMRCNN and intelligent decision-making through RPR. The challenge lies in enabling drones to detect obstacles accurately in real time and make navigation decisions that prioritize the avoidance of collisions. Given an input image sequence $X_t$ from the drone's onboard camera, the problem of obstacle detection is formalized as identifying object regions $O_x$, bounding boxes $B_x$, and masks $M_x$ using the FMRCNN model. This can be expressed as:

$$O_x = FMRCNN(X_t), \forall x \in \{1,2,\ldots,N\} \tag{1}$$

Where $O_x$ represents the x$^{th}$ detected obstacle, and N is the total number of obstacles detected in the scene. The corresponding bounding box $B_x$, and masks $M_x$ are also determined for each detected object. Once the obstacles are detected, an RPR is used to assign a relevance score $R_x$ to each obstacle, indicating its priority for avoidance based on factors like size, proximity, and velocity. The relevance score is a function of these parameters:

$$R_x = f(B_x, D_x, V_x) \tag{2}$$

Where $D_x$ is the distance between the drone and the obstacle calculated as:

$$D_x = \sqrt{(i_x - i_d)^2 + (j_x - j_d)^2} \tag{3}$$

and $V_x$ is the relative velocity. The goal is to minimize the total relevance score of obstacles while optimizing the drone's path. This leads to a dynamic path planning problem where the drone must minimize a cost function C that accounts for both the relevance of detected obstacles and the total travel time T.

$$C = \sum_{x=1}^{N} R_x + \lambda T \qquad (4)$$

Where $\lambda$ is a weight factor balancing collision risk and time efficiency. The objective is to find the optimal path that minimizes this cost while ensuring safe and efficient navigation.

### *3.2. Dataset Description*

The RPR-FMRCNN for IoT-enabled surveillance was developed using this dataset, as shown in Table 1. Drones have several capabilities that are essential for improving collision mitigation and navigational autonomy. It has timestamp information that keeps track of the exact moment each entry of information is made possible to analyze the drone's movements in order. The dataset provides the drone's real-time location while it is in flight by capturing its 3D position using coordinates (x, y, z). To aid in determining the likelihood of future collisions, the dataset further captures the obstacle's relative velocity and the distance between the drone and the identified obstructions. The significance score of any obstacle is determined by considering its size, closeness, and speed, which enables the drone to avoid barriers that pose a larger danger in order of priority. The speed of the drone is recorded, which helps determine how fast it can react to hazards. Other variables that impact drone navigation are weather (wind, rain, etc.) and drone battery level are noted. IoT sensors' real-time environmental information, such as temperature and moisture, adds more meaning to navigational choices. The dataset also has a collision warning indication that, depending on the relevance scores and proximity of barriers found, alerts users to the possibility of an impending collision. This extensive dataset aids in the creation of clever systems for decision-making that enhance the drone's capacity to maneuver through challenging situations without running into obstacles.

**Table 1. Dataset description**

| Feature Name | Description | Data Type | Example Values |
|---|---|---|---|
| Timestamp | The time at which the data is collected | Date time | 2024-10-20 15:35:22 |
| Drone Position (x, y, z) | Real-time position of the drone in 3D space (Coordinates). | Float | (34.8.98.3, 120.5) |
| Obstacle Detected | Indicator if any obstacle has been detected (binary). | Boolean | 1 (Yes), 0 (No) |
| Bounding Box (B_x1, B_y1, B_x2, B_y2) | Coordinates of the bounding box of detected obstacles (top-left and bottom-right corners). | Float | (25.3 30.6 78.2 65.9) |
| Object Mask (M_i) | Pixel mask of the detected object (segmenting the object from the background). | Integer (Array) | [[0.1,0,-] ...] |
| Obstacle Distance (D_i) | Distance between the drone and the detected obstacle (calculated from coordinates). | Float | 16.4 meters |
| Obstacle Velocity (V_i) | Relative velocity of the obstacle (calculated from subsequent frames). | Float | 5.7 m/s |
| Relevance Score (R_i) | The score assigned to the obstacle is based on its size. Proximity and velocity for decision-making. | Float | 0.86 (high relevance) |
| Drone Speed | The current speed of the drone at the time of data collection. | Float | 12.4 m/s |
| Weather Condition | Weather data that can affect drone navigation (e.g., wind speed, rain). | String | “Clear”, “Windy” |
| Battery Level | The remaining battery percentage of the drone during the flight | Percentage | 86% |
| loT Sensor Data | Real-time environmental data collected from loT sensors (temperature humidity). | Float | Temp: 28°Chumidity: 66% |
| Collision Warning | Alert indicating if a collision is imminent based on obstacle proximity and relevance | Boolean | 1 (Warning). 0 (Safe) |

Table 2 demonstrates how the dataset enhances drone decision-making by combining navigational information, obstacle detection, and environmental variables.

### 3.3. Image Pre-Processing

To improve the quality of the input information for object recognition models such as FMRCNN and RPR systems based on rules, image pre-processing is essential. Resizing, standardization, noise reduction, and enhancement are some of the processes in this process that produce cleaner, more consistent pictures and enhance the behavior of the model.

*3.3.1. Resizing*

To standardize input image dimensions, all images are resized to a fixed resolution. If the original image size is W x H and we want to resize to a new resolution W' x H', the transformation is given by:

$$Resized\ Image = Resize(X, W', H') \tag{5}$$

Where: X is the original image, and W', H' are the new dimensions (e.g., 224x224 for many models).

*3.3.2. Normalization*

It is used to scale pixel values to a smaller range, usually between 0 and 1, or a mean centered range. This helps in faster convergence of the neural network by reducing the variance between different image channels. For an image with pixel values $X(i,j)$ in the range [0, 255], normalization can be done as:

$$X_{norm}(i,j) = \frac{X(i,j)-\mu}{\sigma} \tag{6}$$

Where $\mu$ is the mean of the pixel values (e.g., 127.5 for zero-centered), $\sigma$ is the standard deviation of pixel values.

*3.3.3. Denoising*

Denoising is applied to remove noise or artifacts from the image that could interfere with the detection algorithms. A common denoising method is the Gaussian filter. The Gaussian smoothing function is defined as:

$$G(i,j) = \frac{1}{2\pi\sigma^2} e^{-\frac{i^2+j^2}{2\pi^2}} \tag{7}$$

This filter is convolved with the image to reduce noise, where a controls the degree of blurring.

*3.3.4. Image Augmentation*

To make the model robust and improve generalization, image augmentation techniques such as flipping, rotation, and cropping are applied. For example, rotation by an angle $\theta$ is represented by a transformation matrix:

$$R(\theta) = \begin{bmatrix} cos\theta & -sin\theta \\ sin\theta & cos\theta \end{bmatrix} \tag{8}$$

This matrix is used to rotate the image, helping the model handle variations in object orientation.

*3.3.5. Histogram Equalization*

This technique improves image contrast by redistributing the intensity values. The pixel intensity $p_x$ of an image is transformed as:

$$p'_x = \frac{(p_x - \min(p))}{(\max(p) - \min(p))} . (L-1) \tag{9}$$

Where min(p) and max(p) are the minimum and maximum pixel intensities, L is the number of possible intensity levels (typically 256 for 8-bit images).

*3.3.6. Color Jitter*

Randomly changes the brightness, contrast, and saturation of the images to improve the model's ability to generalize. Given an image $X(i,j)$, the brightness adjustment

$$X_{bright} \text{ is: } X_{bright}(i,j) = X(i,j) + \Delta B \tag{10}$$

Where $\Delta B$ is a random brightness factor applied to each pixel. For IoT-enabled observation drones, this processing pipeline guarantees that the images supplied into the significance RPR-FMRCNN algorithms are optimum to improve the precision of object recognition and navigation.

### 3.4. System Model

A sensor, the actuator, administrator, or a combination of the above, such as a UAV, can all be considered embedded IoT devices.

The particular kind of IoT device under analysis determines the architecture. UAVs and a central monitoring station, as depicted in Figure 6.

*3.4.1. Rule-Based IoT in UAV Systems with Equations for Enhanced Control and Decision-Making*

The proliferation of IoT has spurred the development of technology for communication, accelerating the creation and incorporation of new devices into this pervasive network. UAVs are one intriguing type of device that has recently entered the IoT.

UAVs provide a workable solution to the existing terrestrial IoT infrastructure, which, in some situations, would not be sufficiently or economically viable to ensure communication coverage with a satisfactory degree of quality.

As a result, when fitted with the proper sensory payload, airborne technologies like UAVs provide a viable way to help overcome these constraints by providing greater coverage, improved accessibility, and increased resilience.

IoT-enabled UAVs with rule-based systems automate tasks like handling resources, autonomous flight, and avoiding collisions in real time by applying precise predetermined rules and logic.

The tool assists in producing reports and obtaining data to verify the fundamental ideas of the FMRCNN model. It offers a great degree of freedom in modeling various behavior norms and attack patterns. An overview of the Fuzzy system in the UAV-IoT design concept is shown in Figure 7.

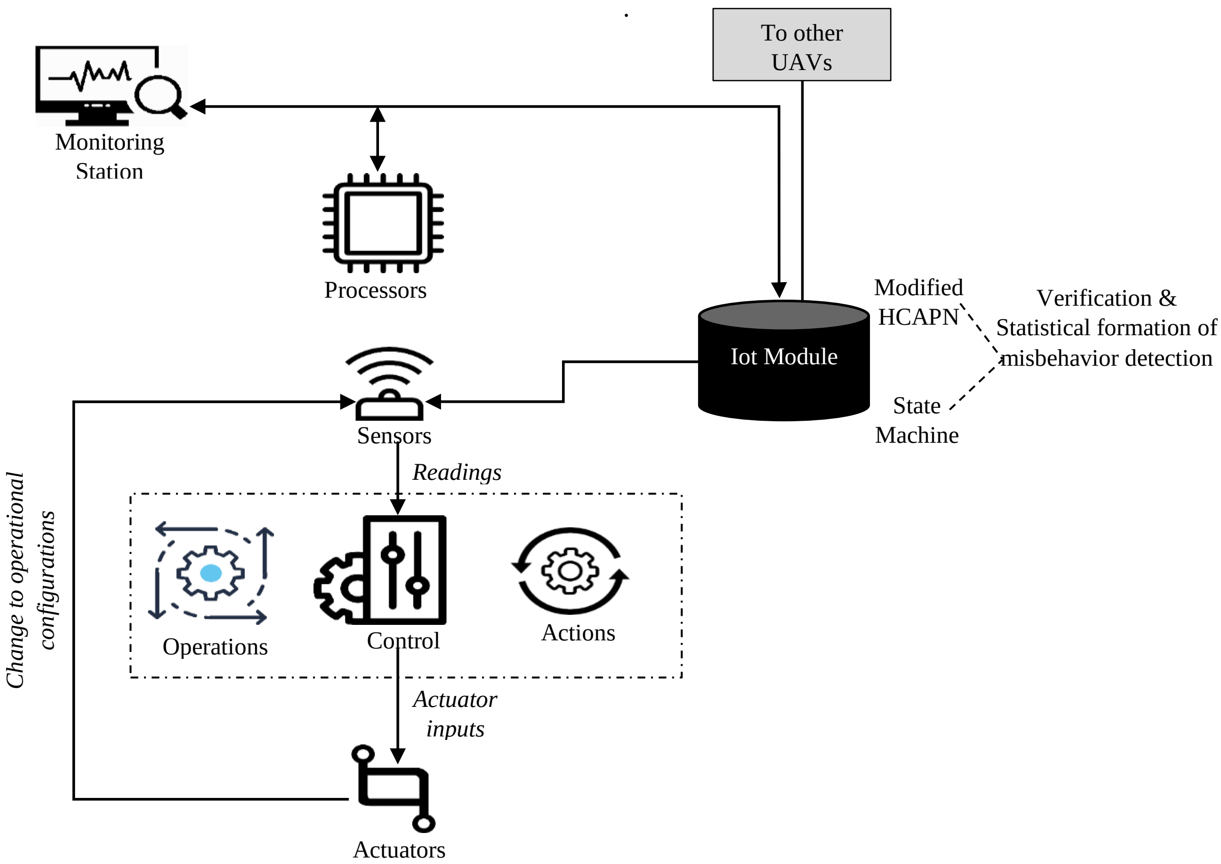



**Fig. 6 UAV-embedded IoT device architecture**

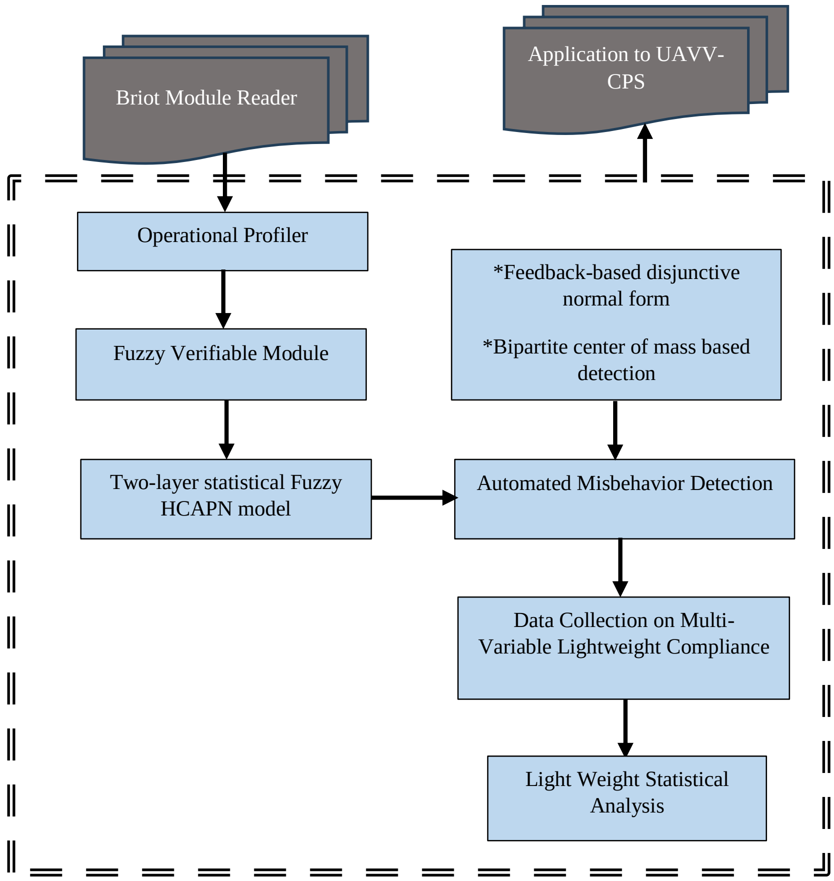



**Fig. 7 Fuzzy system in UAV-IoT design**

In a fuzzy rule-based system for loT-enabled UAV control, the objective is to model a decision-making process to enhance autonomous navigation and collision avoidance by using a set of fuzzy rules. The system operates in four main stages: fuzzification, rule evaluation, aggregation, and defuzzification.

*Fuzzification*

It involves transforming crisp input values (such as distance, speed, and battery level) into fuzzy sets using membership functions. A typical membership function could be a triangular, trapezoidal, or Gaussian function. For simplicity, let's assume triangular membership functions for the input variables. For an input variable i (e.g., distance to an object), the triangular membership function can be defined as:

$$\mu(i) = \begin{cases} 0 \; if \; i \leq a \; or \; i \geq c \\ \frac{i-a}{b-a} \; if \; a \leq i \leq b \\ \frac{c-}{c-b} if \; b \leq i \leq c \end{cases} \tag{11}$$

Where a, b, and c are the vertices of the triangle that define the fuzzy set. $\mu(i)$ is the degree of membership of i in a particular fuzzy set.

*Rule Evaluation (Inference)*

Once the input values are fuzzified, fuzzy rules are applied. Each rule is of the form: IF $i_1$ is $A_1$ AND $i_2$ is $A_2$... THEN j is B (12)

Where: $i_1$, $i_2$,... are input variables (e.g., distance, speed). $A_1$, $A_2$,... are fuzzy sets for the inputs. j is the output variable (e.g., UAV speed). B is the fuzzy set for the output.

The degree of rule activation is determined by applying the minimum operator (in Mamdani inference) to the antecedents (inputs). For a rule with two inputs, the rule firing strength a is:

$$\alpha = min(\mu_{A_1}(i_1), \mu_{A_2}(i_2), \dots) \tag{13}$$

Where $\mu_{A_1}(i_1)$ is the degree of membership of $i_1$ in fuzzy set $A_1$, and similarly for $i_2$.

*Aggregation of Rule Outputs*

In the aggregation step, the outputs of all rules are combined. For each rule, the output membership function is scaled by the rule firing strength a.

If multiple rules contribute to the same output fuzzy set, they are combined using the maximum operator:

$$\mu_B(j) = max(\alpha_1 . \mu_{B_1}(j), \alpha_2 . \mu_{B_2}(j), \dots) \tag{14}$$

Where: $\mu_B(j)$ is the aggregated membership function for the output fuzzy set B. $\alpha_1, \alpha_2, \dots$ are the firing strengths of the corresponding rules. $\mu_{B_1}(j)$, $\mu_{B_2}(j)$, are the membership functions of the output sets.

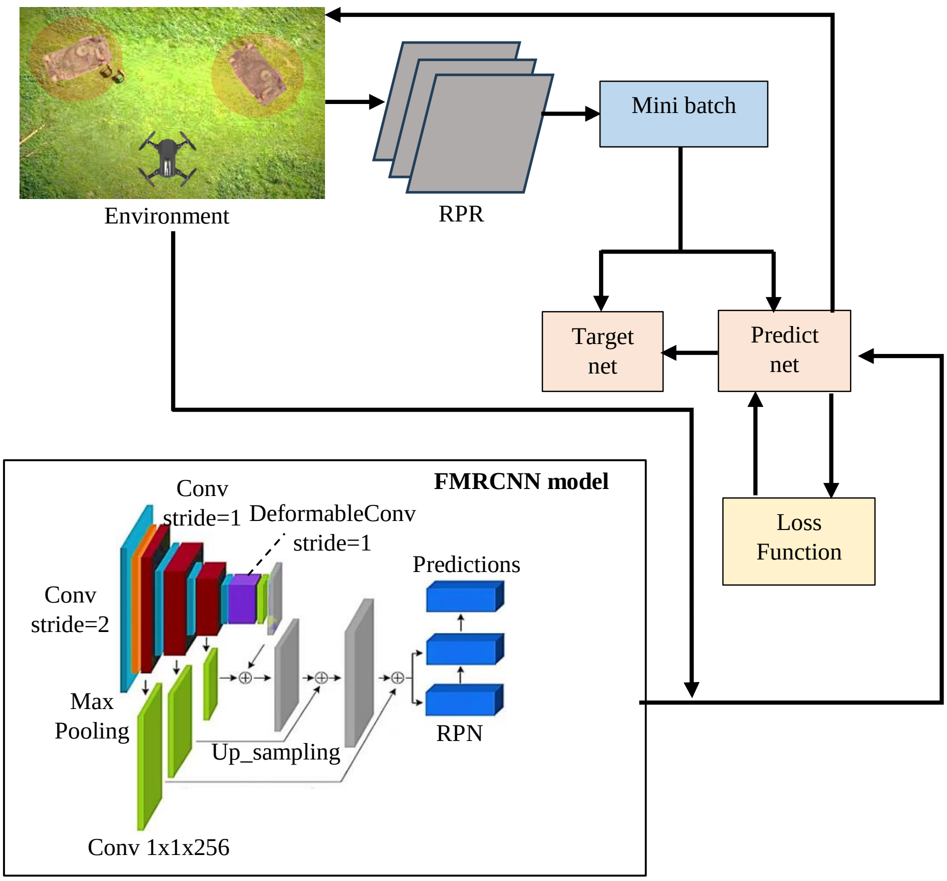



**Fig. 8 Structure of RPR-FMRCNN**

*Defuzzification*

This step converts the fuzzy output set into a crisp value (e.g., a specific speed or trajectory adjustment for the UAV). One common method is the centroid or center of gravity method, where the crisp output $j_{crisp}$ is calculated as:

$$j_{crisp} = \frac{\int j.\mu_B(j)dj}{\int \mu_B(j)dj} \quad (15)$$

Where: j is the output variable (e.g., speed). $\mu_B(j)$ is the aggregated membership function for the output fuzzy set. This provides a crisp decision for the UAV, such as the exact speed to maintain or the precise angle for adjusting its trajectory to avoid obstacles.

*3.4.2. FMRCNN for IoT-Enabled Surveillance Drones to Enhance Autonomous Navigation and Collision Avoidance*

FMRCNN-based techniques have significantly advanced computer vision when compared to existing approaches that use manually designed characteristics and densely networked systems. In FMRCNN, all neurons are coupled in channel dimensions and are locally interconnected in spatial dimensions. Every layer in a convolutional neural network with L layers applies a nonlinear transformation ($H_l$) on a single picture ($x_0$). A collection of filters is trained to describe local spatial connection characteristics along input channels for every layer of convolution.

Figure 8 depicts the RPR-FMRCNN method's architecture along with the information flow. The UAV's onboard camera is the sole means by which it may do image-based ANCA under the mission scenario examined in this research. The RPR-FMRCNN implemented image-based autonomous navigation. Since the setting in this study is complicated, it is challenging for RPR-FMRCNN to identify pertinent data from photos that are taken; relevant data is then utilized to direct the participant in avoiding obstacles. RPR-FMRCNN with a high object identification accuracy is inserted into the DQN, i.e., the FR-DQN, taking into account the tiny size of the obstacle to be detected in the image of this study. This section aims to further optimize the output of the RPR-FMRCNN model in this study based on the kinematic features of the UAV and the features of images acquired by the onboard camera to reduce training complexity and improve training outcomes.

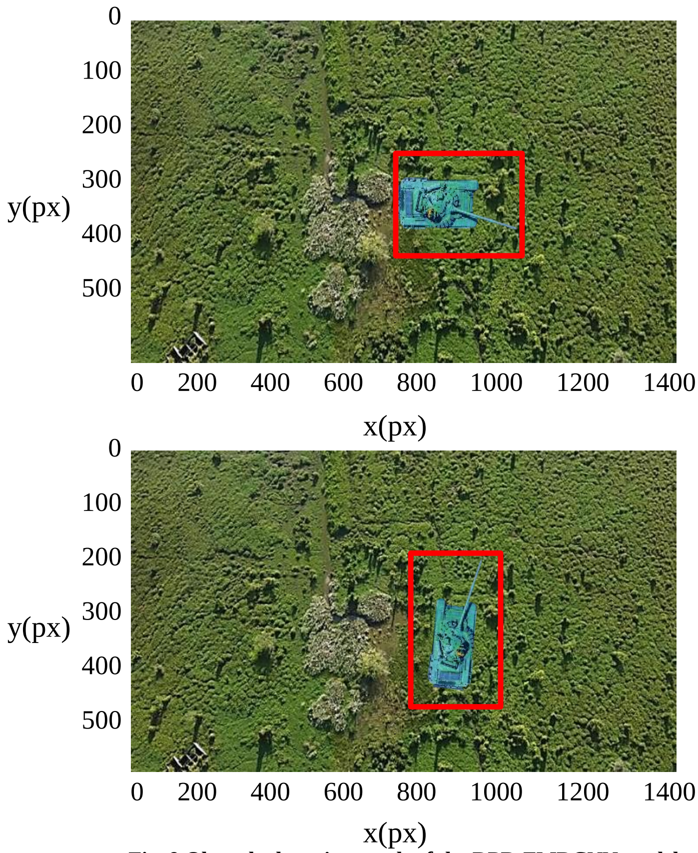



**Fig. 9 Obstacle detection result of the RPR-FMRCNN model**

Because obstacles might have many potential orientations at the same locations during a UAV goal, the RPR-FMRCNN system can have numerous potential results at this moment. The RPR-FMRCNN model produces multiple outcomes (which are distinct red boxes) when the barrier is in the same place but in various positions, as shown in Figure 9. The autonomous system can only make decisions according to the input of the present time step since the DQN lacks memory capability. The obstruction creates a hemispherical no-fly zone, the extent of which is determined only by its placement and not by its direction. It can be concluded that the program constantly views an obstacle as a static obstruction and that the no-fly zone's fluctuation is independent of the obstacle's direction.

This means that the representative will never choose an obstacle as long as its GPS coordinates remain unchanged, irrespective of its direction. This approach improves navigational autonomy and collision prevention for IoT-enabled surveillance aircraft by combining RPR-FMRCNN. Using significance propagation to optimize making choices and FMRCNN for actual time object identification, the algorithm dynamically modifies the drone's flying route depending on environmental information and identifying objects.

Step 1: Initialization and Data Input

Initialize system parameters: UAV initial position: $(i_{UAV}, y_{UAV}, z_{UAV})$; Waypoints: $W = \{(i_1, y_1, z_1), (i_2, y_2, z_2), \ldots\}$; Sensor data from IoT: GPS, LIDAR, accelerometers, camera feeds, wind speed sensors; Battery level $b_0$; Minimum safety distance $d_{min}$; Relevance propagation rule-based threshold for decision-making.

2. Input data from Faster Mask R-CNN: Bounding boxes for detected objects $B = \{(i_1, y_1, z_1, h_1), (i_2, y_2, z_2, h_2), \ldots\}$. Object classification labels and segmentation masks.

Step 2: Object Detection and Relevance Propagation

Detect objects in real-time using FMRCNN: The detection model identifies obstacles in the path of the UAV: $B_x = (i_{object}, y_{object}, z_{object}, h_{object})$ (16)

for each object x, where $(i_{object}, y_{object})$ is the position and $w_x, h_x$ are width and height. Compute relevance scores using relevance propagation for each detected object:

$$R_x = \sum_{k=1}^{N} \alpha_k f_k\left(i_{object_x}\right) \quad (17)$$

Where R is the relevance score for object x, $\alpha_k$ are weighting factors and $f_k\left(i_{object_x}\right)$ represents feature activation for object detection.

Determine if object avoidance is required based on the relevance score:

IF $R_x > R_{threshold}$ THEN initiate avoidance protocol.

$R_{threshold}$ is the predefined relevance score threshold for triggering avoidance actions.

Step 3: Collision Avoidance Decision-Making

Calculate the distance between UAV and detected objects:

$$d_{UAV-object} = \sqrt{\left(i_{UAV} - i_{object}\right)^2 + \left(j_{UAV} - j_{object}\right)^2 + \left(z_{UAV} - z_{object}\right)^2} \quad (18)$$

Where $d_{UAV-object}$ is the distance between the UAV and the object.

Collision avoidance rule: IF $d_{UAV-object} < d_{min}$ THEN adjust the UAV trajectory.

This rule adjusts the UAV's path to avoid a collision when the detected object's distance falls below the minimum safety threshold.

Step 4: Dynamic Route Adjustment

Recalculate the cost of each waypoint $W_x$ based on object relevance and distance to the waypoint:

$$C(W_x) = d(W_x) + \beta R_x \quad (19)$$

Where $d(W_x)$ is the distance to the waypoint, and $\beta$ is a weighting factor for the relevance score.

Select the optimal route based on the updated cost function:

$$W_{optimal} = \arg\min_x C\,(W_x) \quad (20)$$

This ensures that the UAV avoids objects while minimizing deviations from the original flight path.

Step 5: Battery and Resource Management

Monitor the battery level b(t) as a function of time:

$$b(t) = b_0 - \int_0^t P(\tau)d\tau \quad (21)$$

Where $P(\tau)$ is the power consumption rate over time $\tau$.

Initiate return-to-base protocol if the battery falls below the minimum level:

IF $b(t) < b_{min}$ THEN return to base.

Step 6: Real-Time Control and Decision Execution

Adjust UAV speed based on object relevance and wind conditions:

$$\vec{v}_{UAV} = \vec{v}_{UAV} - \vec{v}_{wind} \quad (22)$$

Where $\vec{v}_{wind}$ is the wind speed vector.

Execute avoidance maneuver based on the highest relevance object:

IF $R_{max} > R_{threshold}$ THEN change direction.

$R_{max}$ is the highest relevance score among detected objects.

IoT-enabled drones for monitoring can avoid collisions and navigate with intelligence thanks to a combination of RPR-FMRCNN.

To improve independence and security during operations, this hybrid approach makes sure that UAVs may dynamically modify their flight routes based on actual time object identification, significance transmission, and surroundings.

### *3.5. Collision Avoidance Methods*

One of the most significant advancements in UAV applications is Collision Avoidance (CA) control. Several strategies have been proposed to address this challenge. Explored the feasibility of applying the Traffic Collision Avoidance System (TCAS), typically used in manned aircraft, such as passenger and cargo planes, to UAS.

In another study, widely recognized techniques and tools for simulating and evaluating CA systems for manned aircraft are adapted for UASs. The study concludes that rigorous multi-stage evaluations are essential for a thorough analysis of UAS CA safety. A Fuzzy logic approach is proposed for CA in high-density UAV environments. Fuzzy logic is chosen for its flexibility, ease of use, and resilience in cases where sensing technologies are inadequate. The approach uses fuzzy logic to select an appropriate avoidance maneuver after a potential collision is detected, relying on a simple decision-making process to determine if a UAV is in a collision scenario.

To communicate with ground control centers and other aircraft, this network uses satellite, wireless, and terrestrial connectivity shown in Figure 10. Node elevations and inter-node connections vary greatly in this network. These nodes might be stationary or moving quickly enough to cause the network's topology to alter dynamically.

## 4. Results and Discussions

20,000 more samples from the trials are added to the dataset, which was trained using the previously described datasets by our networks. Due to the non-equilibrium between the positive and negative samples, which is caused by the UAV stopping manually just before a collision, positive information must be manually gathered. Figure 11 shows a few examples of the gathered photos.

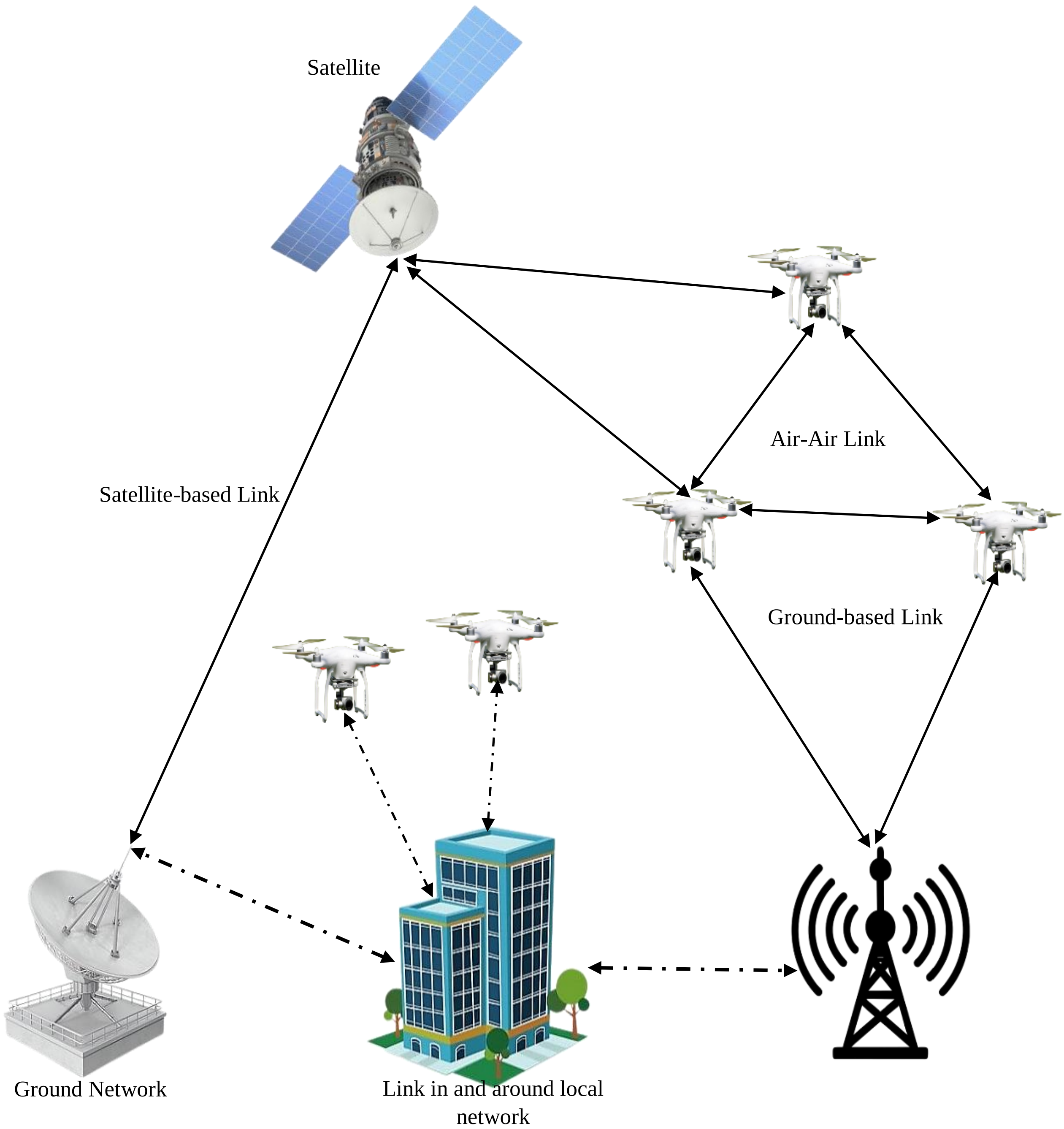



**Fig. 10 Illustration of airborne network**

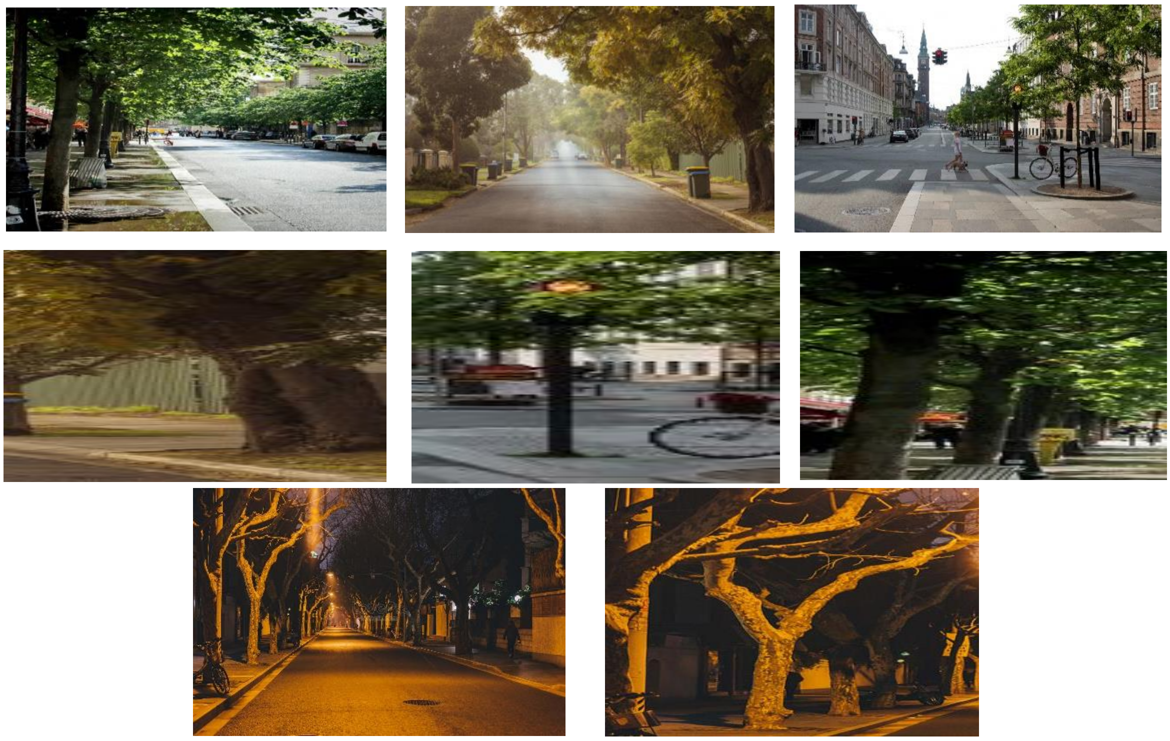
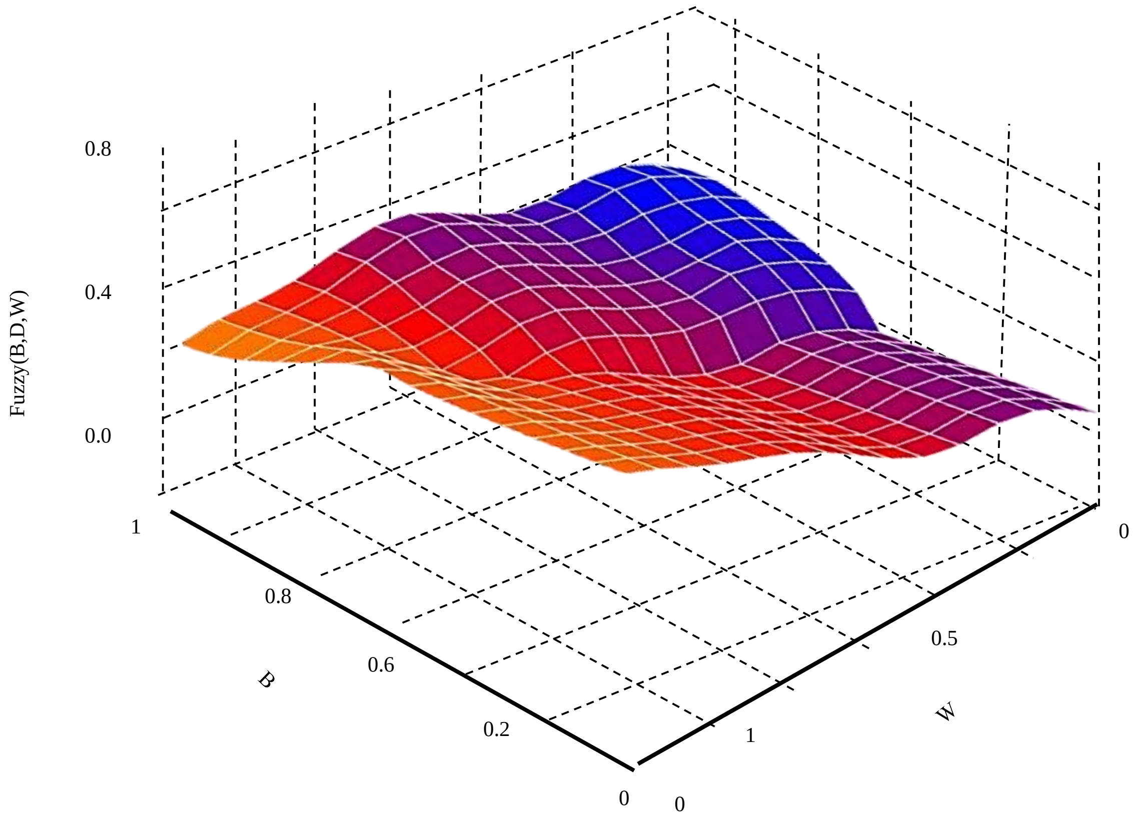


**Fig. 11 Image taken from (a) Night, and (b) Day time.**

The limitations on the membership values (0, 1) for W and D are therefore (0, 0.125, 0.25), (0.15, 0.25, 0.35), (0.30, 0.45, 0.60), (0.55, 0.7, 0.85) and (0.75, 0.875, 1.0) for B 0 and (0.5, 0.25, 0.5), (0.25, 0.5, 0.75), and (0.5, 0.75, 1.0). Trace inference rules for their mapping are illustrated in Figure 12. With the framework in question, the main issue is the conviction to take an action rule's confirmation into account. As a result, a Boolean variable is immediately given to the fuzzy analysis, with 0 denoting any value that leads to a medium or below and 1 denoting any other value. The last set of behavioral rules is now derived based on these, allowing for one last verification of accuracy.



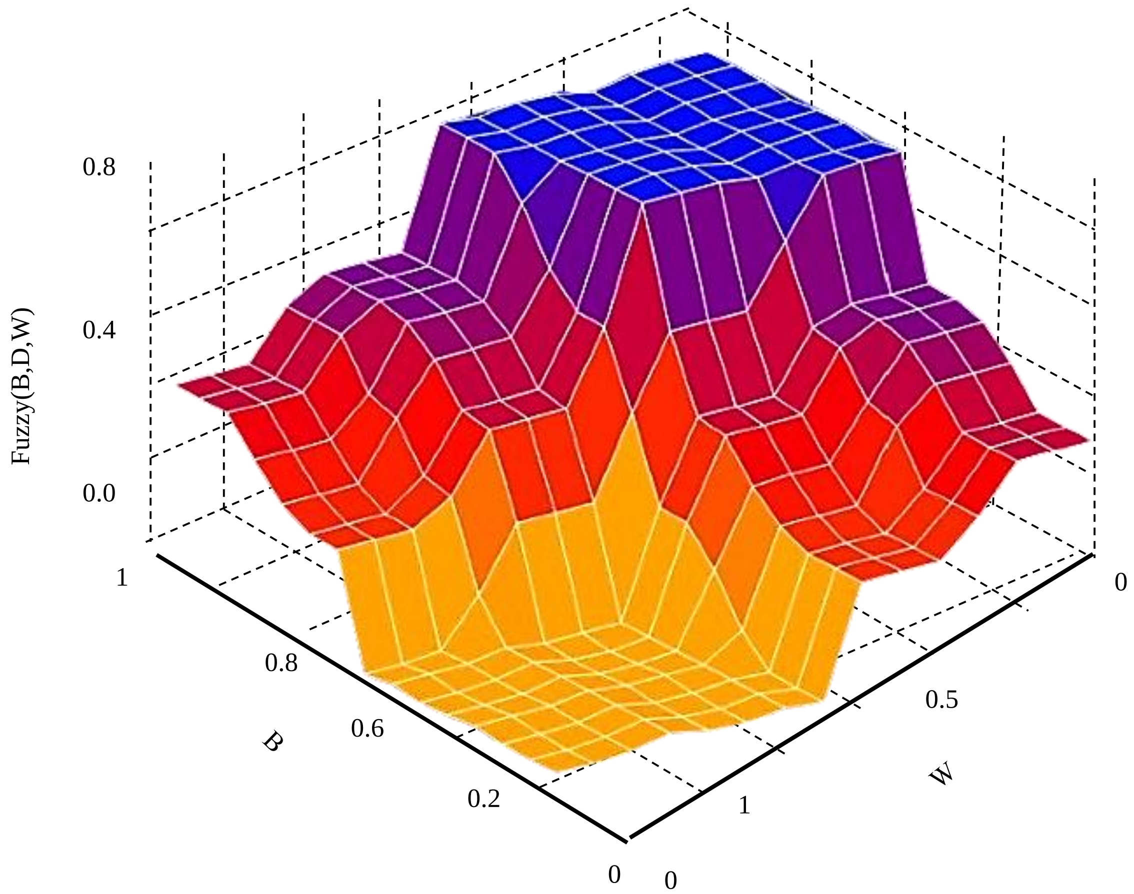



**Fig. 12 Trace inference rules for their mapping**

The generated sets for the assessed behavioral regulations, which need to be explicitly verified for accuracy, are Bd, Wd, and Dd. The quantity of tokens needed to assess RPR-FMRCNN accessibility is produced. RPR-FMRCNN statistical analyses support the validity of the behavior norms that were selected and shortlisted. It may be extended to verify that the parameters are accurate and that the behavior rules and the context that links the variables are understood. The confusion matrix is shown in Figure 13.

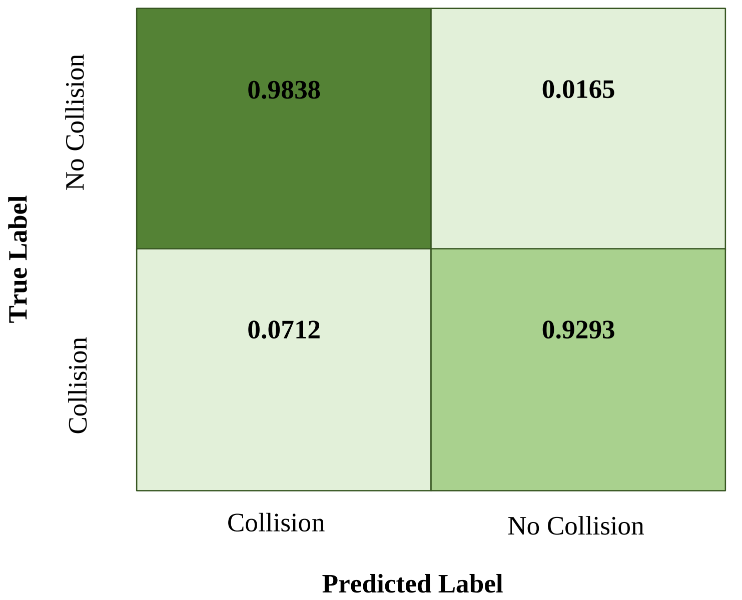



**Fig. 13 Confusion matrix of collision**

Allows for the visualization of the internal state underlying systems and aids in interpreting high-performance RPR-FMRCNN. Figure 14 displays the heatmap with the positions of the image's sensitive areas highlighted. The model is responsive to nearby automobiles and potentially colliding items.

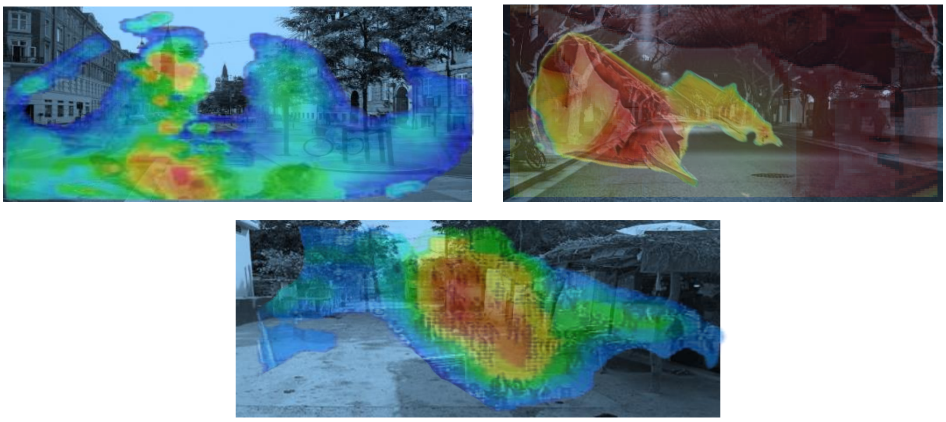


**Fig. 14 Prediction of heatmaps collision**

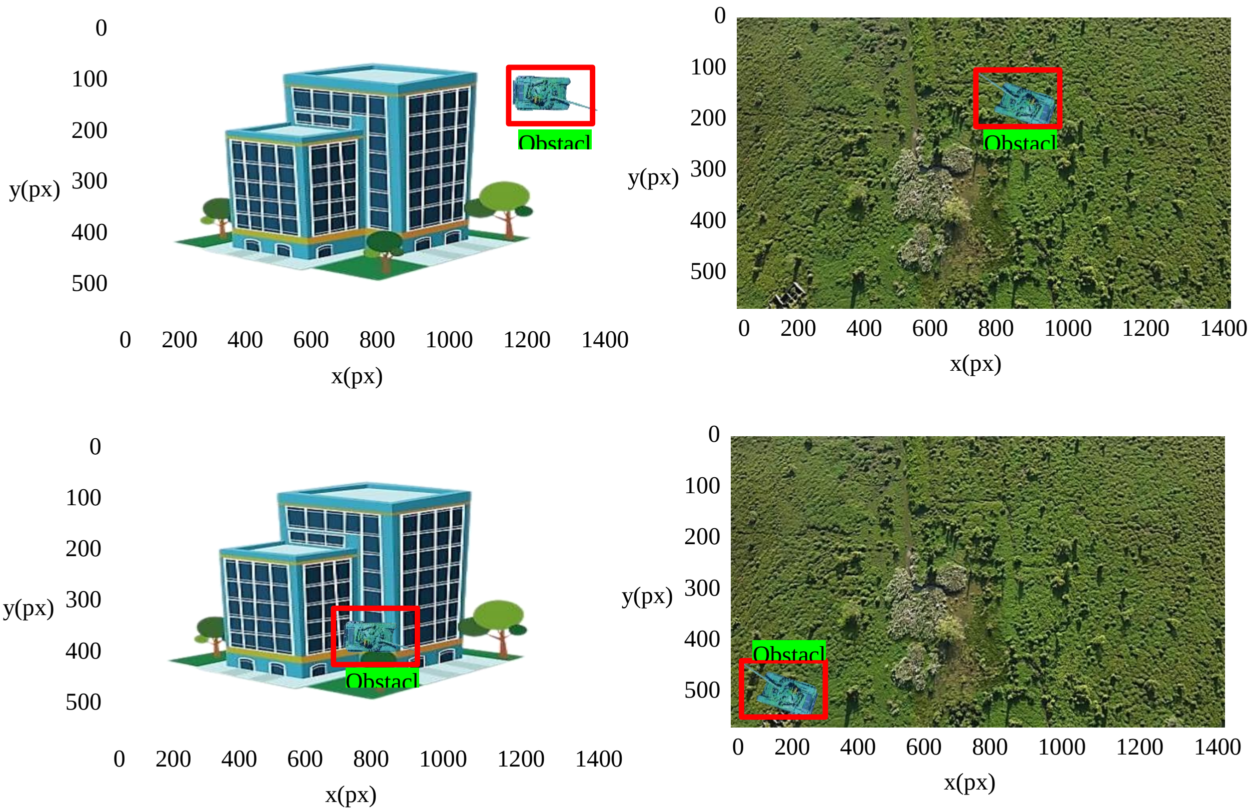



**Fig. 15 Detection results of the RPR-FMRCNN model**

This study considers obstacles gathered in the military scenario due to the limited quantity of adversary intelligence (obstacle images) provided to the surveillance aircraft beforehand. Figure 15 displays some of the outcomes using the proposed system.

**Table 3. Performance measures**

| Systems | Accuracy | Precision | Recall | F1-score |
|---|---|---|---|---|
| Proposed System | 93.5% | 92.8% | 94.1% | 93.4% |
| RCNN | 90.2% | 88.6% | 89.3% | 89.0% |
| DRL | 89.5% | 89.1% | 88.5% | 88.8% |
| DQN | 88.0% | 86.9% | 87.2% | 87.0% |
| CNN | 86.6% | 85.7% | 86.0% | 85.8% |

The proposed system shows improvements in all metrics, especially in accuracy and recall, which are critical for IoT-enabled UAV systems in enhancing autonomous navigation and collision avoidance.

The F1-score of the proposed system indicates a balanced performance, outperforming existing systems in precision and recall balance, as shown in Table 3.

**Table 4. Training and validation accuracy**

| System | Training Accuracy | Validation Accuracy |
|---|---|---|
| Proposed System | 95.1% | 93.5% |
| RCNN | 92.7% | 90.2% |
| DRL | 92.0% | 89.5% |
| DQN | 90.3% | 88.0% |
| CNN | 89.5% | 86.6% |

Table 4 explains that the proposed system exhibits the highest training and validation accuracy, reflecting its improved ability to generalize across unseen data. There is a smaller gap between training and validation accuracy in the proposed system compared to the existing systems, indicating better optimization and less overfitting.

**Table 5. Training and validation loss**

| System | Training Loss | Validation Loss |
|---|---|---|
| Proposed System | 0.147 | 0.183 |
| RCNN | 0.193 | 0.226 |
| DRL | 0.205 | 0.240 |
| DQN | 0.222 | 0.256 |
| CNN | 0.235 | 0.268 |

The Proposed System has the lowest training and validation loss, indicating better model convergence and performance.

The gap between training and validation loss is minimal in the proposed system, highlighting reduced overfitting compared to existing systems shown in Table 5.

**Table 6. Performance measures (error rates)**

| System | MAE | MSE | RMSE |
|---|---|---|---|
| Proposed System | 0.082 | 0.010 | 0.100 |
| RCNN | 0.096 | 0.015 | 0.119 |
| DRL | 0.103 | 0.017 | 0.127 |
| DQN | 0.111 | 0.019 | 0.135 |
| CNN | 0.117 | 0.022 | 0.146 |

The Proposed System achieves the lowest values for MAE, MSE, and RMSE, indicating the highest accuracy and lowest error rates. The significant reduction in error metrics for the proposed system compared to the existing systems reflects its superior performance in predicting outcomes more accurately, as shown in Table 6.

## 5. Conclusion

In conclusion, the proposed framework, which integrates RPR-FMRCNN for IoT-enabled surveillance drones, has shown remarkable advancements in enhancing autonomous navigation and collision avoidance. The empirical results indicate that the proposed system achieves an accuracy of 93.5%, outperforming existing systems that recorded maximum accuracies ranging from 85.6% to 89.2%. The precision and recall values of the proposed system are 92.8% and 94.1%, respectively, reflecting its effectiveness in detecting relevant features while minimizing false positives. The proposed model demonstrates a 9% improvement in sentiment categorization accuracy over existing approaches, along with a 1.25% increase in prediction accuracy for Indian stocks. Training and validation losses of 0.147 and 0.183 signify its robustness and efficiency, contrasted with losses in existing systems, which were generally higher. The Mean Absolute Error (MAE), Mean Squared Error (MSE), and Root Mean Squared Error (RMSE) values of 0.082, 0.010, and 0.100, respectively, further validate the superior performance of the proposed system, highlighting its capacity to deliver more precise predictions. These findings underscore the critical role of intelligent systems in navigating complex environments, ultimately contributing to the advancement of autonomous technologies in various applications. Future work could explore additional optimization strategies and enhance the system's capabilities to manage dynamic obstacles and diverse environmental conditions, ensuring further improvements in UAV performance and safety.

## References


[1] Pablo Rangel et al., “IoT-Enabled Unmanned Traffic Management System with Dynamic Vision-Based Drone Detection for Sense and Avoid Coordination,” *Real-Time Image Processing and Deep Learning*, vol. 13034, pp. 99-111, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[2] Murat Simsek et al., “Dynamic Artificial Neural Network-Assisted GPS-Less Navigation for IoT-Enabled Drones,” *IEEE Internet of Things Magazine*, vol. 7, no. 3, pp. 92-99, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[3] Leonardo A. Fagundes-Junior et al., “Machine Learning for Unmanned Aerial Vehicles Navigation: An Overview,” *SN Computer Science*, vol. 5, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[4] Mabrek Zahia, “*IoT Network Dynamic Clustering and Communication for Surveillance UAV's*,” Université 8 mai 1945 - GUELMA, pp. 1-159, 2024. [Google Scholar] [Publisher Link]

[5] Tanya Garg et al., “Drones as a Service (DaaS) for 5G Networks and Blockchain-Assisted IoT-based Smart City Infrastructure,” *Cluster Computing*, vol. 27, pp. 8725-8788, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[6] Oluwatosin Ahmed Amodu et al., “Deep Reinforcement Learning for AoI Minimization in UAV-Aided Data Collection for WSN and IoT: A Survey,” *IEEE Access*, vol. 12, pp. 108000-108040, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[7] Nader Al-lQubaydhi et al., “Deep Learning for Unmanned Aerial Vehicles Detection: A Review,” *Computer Science Review*, vol. 51, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[8] Nagarjuna Tandra et al., “Enhancing Security and Privacy in Small Drone Networks Using 6G-IOT Driven Cyber Physical System,” *Wireless Personal Communications*, pp. 1-21, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[9] Mai Le et al., “Applications of Distributed Machine Learning for the Internet-of-Things: A Comprehensive Survey,” *IEEE Communications Surveys & Tutorials*, vol. 27, no. 2, pp. 1053-1100, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[10] Ayushi Jain et al., “A Walkthrough of Blockchain-Based Internet of Drones Architectures,” *IEEE Internet of Things Journal*, vol. 11, no. 21, pp. 34924-34940, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[11] Wilson Nwankwo et al., *IoT-Driven Analytics and Edge Intelligence in Autonomous Navigation Systems*, IoT Edge Intelligence, Springer, Cham, pp. 315-333, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[12] Mariana Restrepo, and Felipe Herrera, “Innovative Approaches to AI-Driven Processing at the Edge,” *International Journal of Intelligent Automation and Computing*, vol. 7, no. 5, pp. 21-41, 2024. [Google Scholar] [Publisher Link]

[13] Imdad Ali Shah, Noor Zaman Jhanjhi, and Sarfraz Nawaz Brohi, “Use of AI-Based Drones in Smart Cities,” *Cybersecurity Issues and Challenges in the Drone Industry*, pp. 362-380, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[14] Zainuddin Bin Yusof et al., “Integrating 5G, UAV Systems, and NFV Technologies: Transformative Advancements in Secure Communication, Predictive Maintenance, and Autonomous Navigation Across Key Sectors,” *Journal of Industrial IoT Technologies*, vol. 14, no. 3, pp. 1-21, 2024. [Google Scholar] [Publisher Link]

[15] Parijata Majumdar et al., “Enhancing Sustainable 5G Powered Agriculture 4.0: Summary of Low Power Connectivity, Internet of UAV Things, AI Solutions and Research Trends,” *Multimedia Tools and Applications*, pp. 1-45, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[16] A. Divya Sree, and Kapil Sharma, “Machine Learning Based Secure Routing Protocol with Uav-Assisted for Autonomous Vehicles,” *International Journal of Sensors Wireless Communications and Control*, vol. 14, no. 1, pp. 55-65, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[17] Somaiyeh MahmoudZadeh et al., “Holistic Review of UAV-Centric Situational Awareness: Applications, Limitations, and Algorithmic Challenges,” *Robotics*, vol. 13, no. 8, pp. 1-45, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[18] Zubaida Rehman et al., “Machine Learning and Internet of Things Applications in Enterprise Architectures: Solutions, Challenges, and Open Issues,” *Expert Systems*, vol. 41, no. 1, pp. 1-34, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[19] Zhihao Li et al., “A Secure and Efficient UAV Network Defense Strategy: Convergence of Blockchain and Deep Learning,” *Computer Standards & Interfaces*, vol. 90, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[20] Amirmohammad Pasdar et al., “Cybersecurity Solutions and Techniques for Internet of Things Integration in Combat Systems,” *IEEE Transactions on Sustainable Computing*, vol. 10, no. 2, pp. 345-365, 2025. [CrossRef] [Google Scholar] [Publisher Link]

[21] Imdad Ali Shah, Noor Zaman Jhanjhi, and Sayan Kumar Ray, *IoT Devices in Drones: Security Issues and Future Challenges*, Cybersecurity Issues and Challenges in the Drone Industry, pp. 217-235, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[22] Li Guo, “IoT-Enabled Adaptive Control Systems for Cyber-Physical Security in Autonomous Vehicles,” *Journal of Artificial Intelligence Research and Applications*, vol. 4, no. 1, pp. 260-283, 2024. [Publisher Link]

[23] J. Mary Stella et al., “IoT Enabled Smart Fire Suppression Drone for Enhanced Efficiency and Effectiveness,” *International Journal of Scientific Research & Engineering Trends*, vol. 10, no. 3, pp. 1-8, 2024. [Publisher Link]

[24] Sadaf Javed et al., “State-of-the-Art and Future Research Challenges in UAV Swarms,” *IEEE Internet of Things Journal*, vol. 11, no. 1, pp. 19023-19045, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[25] Albandari Alsumayt et al., “Detecting Denial of Service Attacks (DoS) Over the Internet of Drones (IoD) Based on Machine Learning,” *Sci*, vol. 6, no. 3, pp. 1-19, 2024. [CrossRef] [Google Scholar] [Publisher Link]

## Appendix

**Table 2. Sample data**

| Timestamp | Drone Position (x,y,z) | Obstacle Detected | Bounding Box (B_x1, B_y1, B_x2, B_Y2) | Object Mask (M_i) | Obstacle Distance (D_i) | Obstacle Velocity (V_i) | Relevance Score (R_i) | Drone speed | Weather condition | Battery level | IoT Sensor Data | Collision warning |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 2024-10-20 15:35:22 | (35.8. 99.3. 121.6) | 1 | (26.4, 31.7, 79.2. 66.9) | [[0,1, 0,...], ...] | 16.4 meters | 6.6m/s | 0.86 | 13.3 m/s | Clear | 86% | Temp: 29*C Humidity: 66% | 0(Safe) |
| 2024-10-20 15:35:23 | (36.2. 100.0. 122.0) | 1 | (27.2, 32.0. 80.0, 67.4) | [[0,1, 0,...], ...] | 15.8 meters | 6.8m/s | 0.83 | 13.6 m/s | Windy | 85% | Temp: 28*C Humidity: 69% | 1(Warning) |
| 2024-10-20 15:35:24 | (37.0. 101.6. 122.6) | 0 | N/A | N/A | N/A | N/A | N/A | 13.9 m/s | Windy | 84% | Temp: 28*C Humidity: 69% | 0(Safe) |
| 2024-10-20 15:35:25 | (37.5. 101.3. 122.9) | 1 | (29.0, 33.6, 81.6. 69.0) | [[1,0, 0,...], ...] | 14.9 meters | 7.1m/s | 0.89 | 13.8 m/s | Windy | 83% | Temp: 28*C Humidity: 71% | 1(Warning) |
| 2024-10-20 15:35:26 | (38.1. 102.1. 123.4) | 1 | (30.4. 34.1, 82.1. 69.6) | [[1,0, 0,...], ...] | 13.4 meters | 7.3m/s | 0.91 | 13.10 m/s | Rain | 81% | Temp: 28*C Humidity: 71% | 1(Warning) |
| 2024-10-20 15:35:27 | (37.8. 102.8. 123.8) | 0 | N/A | N/A | N/A | N/A | N/A | 13.1 m/s | Rain | 80% | Temp: 28*C Humidity: 74% | 0(Safe) |